Finding and drawing contours

Finding object by their shape is one of the key reason people usually comes to OpenCV.

```
(ns lush-storm
  (:require
    [opencv3.utils :as u]
    [opencv3.colors.html :as color]
    [opencv3.core :refer :all]))
```

```
nil
```

Finding contour of headphone and drawing rectangles ...

```
(def headphones
  (imread "resources/morph/headphone.png"))

(def image-c
  (clone headphones))
```

```
#'lush-storm/image-c
```

```
(def contours
  (new-arraylist))

(find-contours
  (-> headphones clone (cvt-color! COLOR_BGR2GRAY))
  contours
  (new-mat)
  RETR_TREE CHAIN_APPROX_SIMPLE)
```

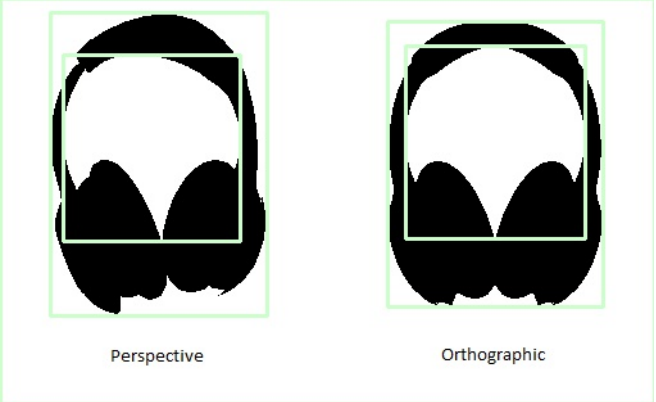
```
nil
```

```
(def interesting-contours
  (filter
    #(and (> (contour-area %) 100 ) (> (.height (bounding-rect %)) 28))
    contours))
```

```
#'lush-storm/interesting-contours
```

```
(doseq [c interesting-contours]
    (let [ rect (bounding-rect c)]
      (rectangle
        image-c
        (new-point (.x rect) (.y rect))
        (new-point (+ (.width rect) (.x rect)) (+ (.y rect) (.height rect)))
        (color/->scalar "#ccffcc")
        2)))

(u/mat-view image-c)
```



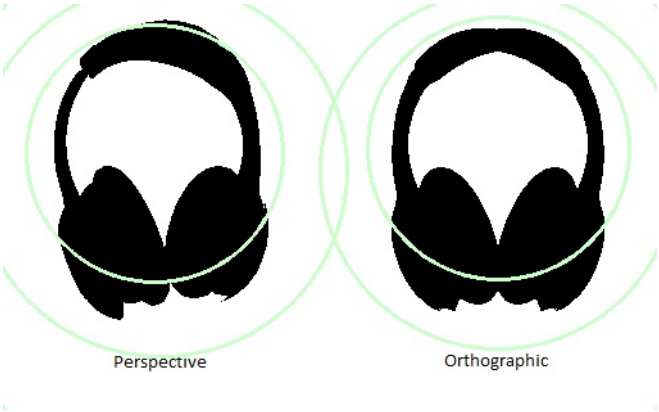
Using core draw-contours to ... draw the contours

```
(def image-d (clone headphones))

(doseq [c interesting-contours]
 (let [ rect (bounding-rect c) center (u/center-of-rect rect) ]
   (circle image-d
           center
           (u/distance-of-two-points center (.tl rect))
           (color/->scalar "#ccffcc")
```

```
            2)))

(u/mat-view image-d)
```




```
(def image-c (clone headphones))

(dotimes [ci (.size contours)]
 (draw-contours
   image-c
   contours
   ci
   (color/->scalar "#ccffcc")
   2))

(u/mat-view image-c)
```